Java GCD Function

Given the following GCD function:

```
public static int gcd(int p, int q) {
    if (q == 0) return p;
    return gcd(q, p % q);
}
```

Computing `gcd(gcd(a, b), gcd(c, d))`

Given four positive integers `a`, `b`, `c`, and `d`, the Java program would compute `gcd(gcd(a, b), gcd(c, d))` as follows:

1. **Step 1: Compute `gcd(a, b)`**
    - Call `gcd(a, b)` to get the GCD of `a` and `b`. Let's denote this result as `g1`.
    - In Java: `int g1 = gcd(a, b);`
2. **Step 2: Compute `gcd(c, d)`**
    - Call `gcd(c, d)` to get the GCD of `c` and `d`. Let's denote this result as `g2`.
    - In Java: `int g2 = gcd(c, d);`
3. **Step 3: Compute `gcd(g1, g2)`**
    - Call `gcd(g1, g2)` to get the GCD of `g1` and `g2`.
    - In Java: `int result = gcd(g1, g2);`

Putting it all together in a Java program:

```
public class GCDExample {
    public static void main(String[] args) {
        int a = 12;  // Example value for a
        int b = 15;  // Example value for b
        int c = 9;   // Example value for c
        int d = 6;   // Example value for d

        int g1 = gcd(a, b);
        int g2 = gcd(c, d);
        int result = gcd(g1, g2);

        StdOut.println("gcd(gcd(" + a + ", " + b + "), gcd(" + c + 
", " + d + ")) = " + result);
    }

    public static int gcd(int p, int q) {
        if (q == 0) return p;
        return gcd(q, p % q);
    }
}
```

Explanation of the Program

1. **Computing `gcd(a, b)`:**

   - The `gcd` function calculates the greatest common divisor of `a` and `b`.
   - For example, if `a = 12` and `b = 15`,
   - Factors of `12`: `1, 2, 3, 4, 6, 12`
   - Factors of `15`: `1, 3, 5, 15`
   - Common factors: `1, 3` `gcd(12, 15)` = `3` `gcd(12, 15)` is calculated as follows:
     - `gcd(12, 15)` calls `gcd(15, 12 % 15)` which is `gcd(15, 12)`
     - `gcd(15, 12)` calls `gcd(12, 15 % 12)` which is `gcd(12, 3)`
     - `gcd(12, 3)` calls `gcd(3, 12 % 3)` which is `gcd(3, 0)`
     - `gcd(3, 0)` returns `3`
   - So, `gcd(12, 15)` is `3`
2. **Computing `gcd(c, d)`:**

   - The `gcd` function calculates the greatest common divisor of `c` and `d`.
   - For example, if `c = 9` and `d = 6`, `gcd(9, 6)` is calculated as follows:
     - `gcd(9, 6)` calls `gcd(6, 9 % 6)` which is `gcd(6, 3)`
     - `gcd(6, 3)` calls `gcd(3, 6 % 3)` which is `gcd(3, 0)`
     - `gcd(3, 0)` returns `3`
   - So, `gcd(9, 6)` is `3`
3. **Computing `gcd(g1, g2)`:**

   - Finally, the program calculates the GCD of `g1` and `g2`.
   - For example, `g1 = 3` and `g2 = 3`, `gcd(3, 3)` is calculated as follows:
     - `gcd(3, 3)` calls `gcd(3, 3 % 3)` which is `gcd(3, 0)`
     - `gcd(3, 0)` returns `3`
   - So, `gcd(3, 3)` is `3`

Output

For the given values `a = 12`, `b = 15`, `c = 9`, and `d = 6`, the output of the program will be:

```
gcd(gcd(12, 15), gcd(9, 6)) = 3
```

This confirms that the value computed by `gcd(gcd(a, b), gcd(c, d))` is the greatest common divisor of all four integers `a`, `b`, `c`, and `d`.